

Испит из Релационих база података, фебруар 2019. год. (И смер), I група

Број индекса	Име и презиме

Задачи се раде 240 минута. Максималан број поена је 70. Број поена на испиту се израчунава тако што се саберу освојени поени по задацима, збир подели са два и заокружи. Број поена по задацима је:

Задатак	1	2	3	4	Збир 1-4	5	6	7	Збир 5-7	Укупно	
Максимално	20	20	30	20	90	Збир/2	12	16	22	50	Збир/2
Освојено											

- Навести и описати функције СУБП-а
 - Дати дефиницију и описати релациони оператор сумаризације
 - Написати SQL упит који за све табеле које је направио корисник а које нису коришћене у последњих годину дана у односу на тренутак када се извршава упит, формира **ниску** која садржи SQL наредбу за брисање тих табела. На пример, ако табеле *dosije* и *ispitni_rok* нису коришћене у последњих годину дана, садржај ниске треба да буде `drop table dosije; drop table ispitni_rok;`
- Написати кориснички дефинисану функцију *Povrsina_Trougla(stranica1, stranica2, ugao)* чији су аргументи дужине две странице троугла и угао између њих, а вредност површина тог троугла. Вредност угла је задата у радијанима. Сва три аргумента су реални бројеви записани помоћу декадне основе. У функцији извршити проверу да ли су аргументи коректно задати, и у случају да нису приказати одговарајућу поруку.
- Нека је дата релациона променљива $R = \{A, B, C, D, E, F, G\}$ и скуп Ф ФЗ:
 - $A \rightarrow D$
 - $ADG \rightarrow F$
 - $ACE \rightarrow BD$
 - $B \rightarrow C$
 - $C \rightarrow A$
 - $D \rightarrow G$
 - $E \rightarrow B$
 - $EF \rightarrow AD$
 - $FG \rightarrow EC$
 - $G \rightarrow F$
 - Одредити нередуцибилни покривач скупа функционалних зависности F.
 - Одредити све кандидате за кључ релације R.
 - Испитати да ли је релација R у BCNF и декомпоновати је ако није.
 - Нека је $R_1\{A, C, D, F\} = \pi\{R\}$. Одредити скуп ФЗ релације R_1 .

Обавезно образложити СВЕ кораке у раду. Навођење само резултата појединих корака неће бити признато као делимично урађен задатак.

- Написати SQL наредбу којом се приказује број студента који су добили неку од оцена за сваку комбинацију предмета, испитних рокова и оцена, као и индикаторе (0/1) који показују који од атрибута је узет у обзир при налажењу броја студената.
 - Нека је поглед *A* дефинисан над погледом *B* који је дефинисан над основном табелом *C*. Навести случајеве у којима може или не може да се ажурира основна табела *C* преко погледа *A*.
 - Нека је дата табела `create table dosije_xml (podatak xml not null)` при чему су изглед и структура слога из табеле

```
<?xml version="1.0" encoding="UTF-8" ?>
<student>
  <broj_indeksa>"20140021"</broj_indeksa>
  <ime>Miloš</ime>
  <prezime>Perić</prezime>
  <datum_rodjenja>"20.01.1995"</datum_rodjenja>
</student>
```

Написати упит који враћа број индекса, име, презиме, и број положених испита просек оцена свих студената рођених 1997 године чији се подаци налазе у табели `dosije_xml`.

5. Написати упит на SQL-у којим се издваја назив испитног рока, број дана рока за пријаву испита за тај рок, назив предмета, име и презиме студента и оцену. За издвојеног студента мора важити да је у том року положио испит из тог предмета који је некада у ранијем року пао, а положио је све условне предмете за тај положени предмет. Сматрати да је пад у ранијем року, ако је датум пријаве тог полагања ранији од датума почетка пријављивања за издвојени рок.
6. Написати упит на SQL-у којим се издваја назив предмета, назив смера и број студената тог смера који су испит из тог предмета положили са 10. Издвојити парове предмет-смер, само ако су сви студенти тог смера некада уписали курс из тог предмета, при чему је морао неки студент тог смера да упише курс из тог предмета. Уколико ниједан студент није добио 10 на испиту из тог предмета издвојити 0. Резултат упита уредити опадајуће по броју студената.
7. По завршетку школске године, на факултету се одређује студент генерације за сваки смер. Студент генерације за једну школску годину и један смер је студент који

- је дипломирао у току те школске године (нпр. за школску 2018/2019 годину, студент је дипломирао између 1.10.2018. и 30.9.2019);
- има просек изнад 8,5;
- има највећу оцену успеха која се рачуна по формули: $prosek_studenta_na_polozenim_ispitima * predvidjena_duzina_studiranja_na_smeru_studenta * 365 / broj_dana_studiranja_studenta$
- Написати SQL наредбу којом се прави табела **student_generacije** која садржи податке о студентима генерације за сваку школску годину и смер. Табела има следеће колоне:
 - *id_smera* - идентификатор смера;
 - *godina* - школска година (пример: за школску годину 2018/2019 чува се вредност 2018);
 - *indeks* - индекс студента;
 - *prosek* - просечна оцена на положеним испитима студента;
 - *duzina_studiranja* - дужина студирања студента изражена у данима;

Дефинисати примарни кључ над табелом **student_generacije**. У једној школској години може бити један или више студената генерације на смеру.

- Написати (једну) SQL наредбу којом се дефинише страни кључ у табели **student_generacije** на табелу *dosije*. Обезбедити да се при брисању студента из табеле *dosije* бришу и подаци тог студента из табеле **student_generacije**. Поставити и ограничење да просек студента мора бити већи од 8,5.
- Написати SQL наредбу којом се у табелу **student_generacije** уносе подаци о студентима генерације за смерове на основним студијама. За студенте генерације унети: идентификатор смера, школску годину и индекс.
- Написати SQL наредбу којом се у табели **student_generacije** ажурирају вредности колоне *prosek* и *duzina_studiranja*.
- Написати SQL наредбу којом се брише табела **student_generacije**.

Упутство:

Направите нови директоријум на Desktop-у и назовите га у формату **rbpFebG1.brojIndeksa.ime.prezime**. Нпр. студент Марко Марковић са бројем индекса 125/2013 треба да направи директоријум *rbpFebG1.2013125.Marko.Markovic*.

У том директоријуму оставите решења задатака. Решење сваког задатка оставите у посебној датотеци. Датотеке са решењима назовите редним бројем задатка (нпр. *1.sql*, *2.sql* ...). **За сваки задатак оставите тачно једно решење, у супротном задатак неће бити прегледан.**

Задаци се решавају над базом података **stud**.

Синтаксно неисправна решења за први, други и трећи задатак носе 0 од предвиђеног броја поена.

Дозвољена литература се налази у директоријуму `/home/ispit/Desktop/Materijali/rbpKnjige`.

Испит из Релационих база података, фебруар 2019. год. (И смер), II група

Број индекса	Име и презиме

Задачи се раде 240 минута. Максималан број поена је 70. Број поена на испиту се израчунава тако што се саберу освојени поени по задацима, збир подели са два и заокружи. Број поена по задацима је:

Задатак	1	2	3	4	Збир 1-4		5	6	7	Збир 5-7		Укупно
Максимално	20	20	30	20	90	Збир/2	12	16	22	50	Збир/2	
Освојено												

- Навести и описати основне компоненте релационог модела
 - Описати својства трансакција у релационим базама
 - Написати SQL упит који за СВЕ табеле које је направио корисник који тренутно извршава упит издваја ниску која садржи SQL упит за издвајање свих атрибута који су експлицитно побројани. На пример, за табелу *ispitni_rok* са атрибутима *godina_roka*, *oznaka_roka*, *naziv* добија се ниска чији је садржај `select naziv, oznaka_roka, godina_roka from ispitni_rok;`
- Написати кориснички дефинисану функцију *Povrsina_Mnogougla(duzina_stranice, broj_stranica)* чији су аргументи дужина странице и број страница многоугла, а вредност површина тог троугла. Дужина странице је реалан број са највише 5 цифаре у целобројном и 2 цифре у разломљеном делу, а број страница је целобројна вредност. У функцији извршити проверу да ли су аргументи коректно задати, и у случају да нису приказати одговарајућу поруку.

3. Нека је дата релациона променљива $R = \{A, B, C, D, E, F, G\}$ и скуп $F \Phi Z$:

- | | |
|-------------------------|------------------------|
| 1) $A \rightarrow D$ | 6) $D \rightarrow G$ |
| 2) $ADG \rightarrow F$ | 7) $EG \rightarrow BC$ |
| 3) $ACE \rightarrow BD$ | 8) $EF \rightarrow AD$ |
| 4) $B \rightarrow C$ | 9) $F \rightarrow E$ |
| 5) $C \rightarrow A$ | 10) $G \rightarrow F$ |

- Одредити нередуцибилни покривач скупа функционалних зависности F .
- Одредити све кандидате за кључ релације R .
- Испитати да ли је релација R у BCNF и декомпоновати је ако није.
- Нека је $R_1\{A, C, D, G\} = \pi\{R\}$. Одредити скуп ΦZ релације R_1 .

Обавезно образложити СВЕ кораке у раду. Навођење само резултата појединих корака неће бити признато као делимично урађен задатак.

- Нека је PS просечна оцена, а VR варијанса оцене сваког од студената. Формирати MQT *Van_intervala* који садржи број индекса, име, презиме, просечну оцену студента и ранг студента према варијанси оцене и издвојити све студенте који су ранжирани на парним местима у списку.
 - Навести услове под којима може да се ажурира поглед у систему Db2.
 - Нека у бази постоји табела `create table dosije_xml (podatak xml not null)` при чему су изглед и структура слога из табеле

```
<?xml version="1.0" encoding="UTF-8" ?>
<student>
  <broj_indeksa>"20140021"</broj_indeksa>
  <ime>Petar</ime>
  <prezime>Petrović</prezime>
  <jmbg>"2001995711013"</jmbg>
</student>
```

Написати упит који враћа број индекса, име, презиме, датум рођења и просек оцена свих студената чији се подаци налазе у табели `dosije_xml`.

5. Написати упит на SQL-у којим се издваја назив смера, индекс, име и презиме свих студената тог смера који су положили неки обавезан предмет на свом смеру који носи више од 4 ЕПСБ. Резултат уредити према презимену студента у растућем и броју индекса опадајућем поретку.

6. Написати упит на SQL-у којим се за сваког дипломираног студента издвајају информације:

- у једној колони, име и презиме студента, колону назвати *Puni potpis*;
- колико дана је студент студирао (број дана од датума уписа на факултет до датума дипломирања, при чему је студент дипломирао онда када његов статус постане 'дипломирао', а уписан је онда када је попуњен његов досије (датум уписа студента налази се у табели *dosije*)) ;
- колико дана је провео пријављивајући испите током студија (информације о укупном трајању, у данима, пријављивања свих испитних рокова у којима је пријављивао испите). Колону назвати *t*.

7. На факултету се додељују новчане награде студентима. Студент добија новчану награду ако задовољава услов: просечна оцена на положеним испитима му је већа од просечне оцене добијене на положеним испитима студената који су на истом смеру као и он и који су факултет уписали исте године као и он.

- Написати SQL наредбу којом се прави табела **nagrada** која садржи податке о наградама студената. Табела има следеће колоне:
 - *indeks* - индекс студента који је добио награду;
 - *godina_upisa* - календарска година уписа тог студента на факултет;
 - *nagrada* - висина награде.
- Написати (једну) SQL наредбу којом се дефинише страни кључ у табели **nagrada** на табелу *dosije*. Обезбедити да се при брисању студента из табеле *dosije* бришу и подаци тог студента из табеле **nagrada**. Поставити и ограничење да година уписа мора бити између 2000. и 2050. и да је подразумевана вредност колоне *nagrada* 10000.
- Написати SQL наредбу којом се у табелу **nagrada** уносе подаци за студенте који су студирали само на терету буџета и нису дипломирали. Висина награде се рачуна по формули: просечан број бодова које је студент добио на положеним испитима * 500.0
- Написати SQL наредбу којом се из табеле **nagrada** бришу подаци о студентима који нису положили све обавезне предмете са смера који студирају.
- Написати SQL наредбу којом се брише табела **nagrada**.

Упутство:

Направите нови директоријум на Desktop-у и назовите га у формату **rbpFebG2.brojIndeksa.ime.prezime**. Нпр. студент Марко Марковић са бројем индекса 125/2013 треба да направи директоријум *rbpFebG2.2013125.Marko.Markovic*.

У том директоријуму оставите решења задатака. Решење сваког задатка оставите у посебној датотеци. Датотеке са решењима назовите редним бројем задатка (нпр. *1.sql, 2.sql ...*). **За сваки задатак оставите тачно једно решење, у супротном задатак неће бити прегледан.**

Задаци се решавају над базом података **vstud**.

Синтаксно неисправна решења за први, други и трећи задатак носе 0 од предвиђеног броја поена.

Дозвољена литература се налази у директоријуму `/home/ispit/Desktop/Materijali/rbpKnjige`.